COMPLAINT EXHIBIT 14
U.S. Patent No. 8,050,321 (H.264)

As demonstrated in the chart below, ASUS directly and indirectly infringes at least claims 8 of U.S. Patent No. 8,050,321 (the "'321 Patent"). ASUS directly infringes, contributes to the infringement of, and/or induces infringement of the '321 Patent by making, using, selling, offering for sale, and/or importing into the United States the Accused Products that are covered by one or more claims of the '321 Patent. The Accused Products are devices that decode H.264-compliant video. For example, the ASUS Q543MV Notebook ("ASUS Q543MV") is a representative product for other ASUS devices that decode H.264-compliant video.

The ASUS Q543MV contains at least one video decoder that helps decode H.264-compliant video.[1] While evidence from the ASUS Q543MV is specifically charted herein, the evidence and contentions charted herein apply equally to the other ASUS Accused Products that decode H.264-compliant video.

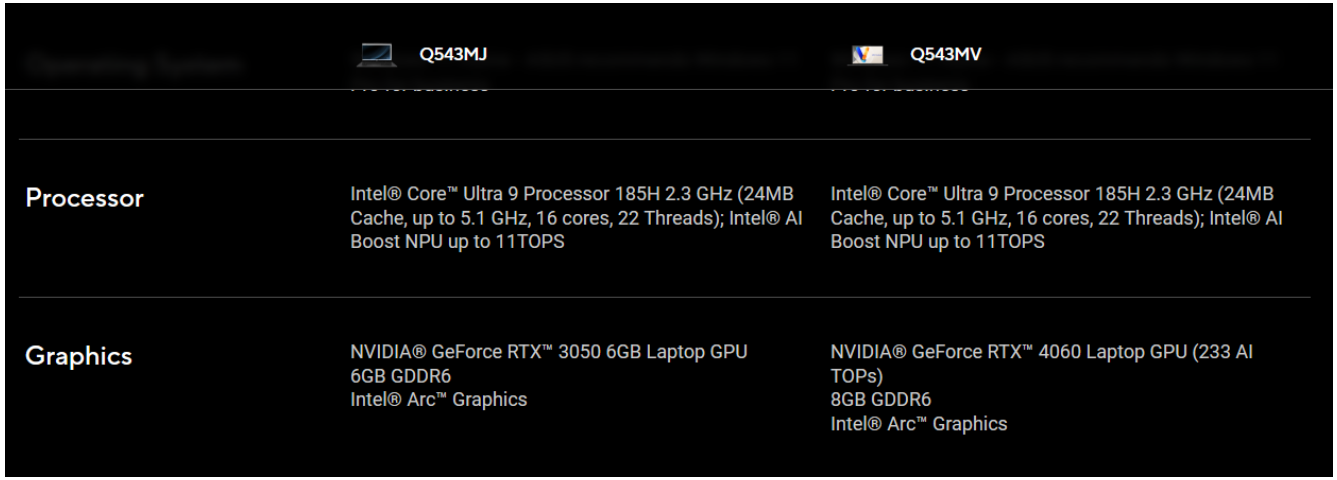
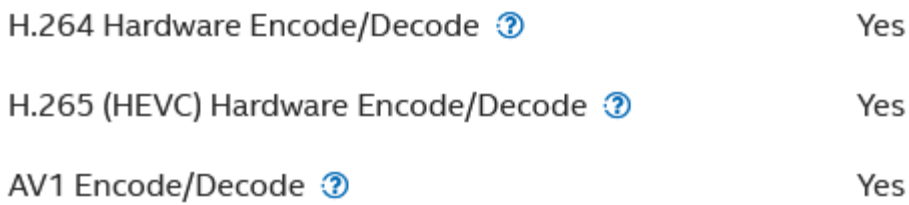
No part of this exemplary chart construes, or is intended to construe, the specification, file history, or claims of the '321 Patent. Moreover, this exemplary chart does not limit, and is not intended to limit, Nokia's infringement positions or contentions.

The following infringement chart includes exemplary citations to ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services (available at https://www.itu.int/rec/T-REC-H.264-200503-S/en) (the "H.264 Standard"). The cited functionality has been included in editions of the H.264 Standard since at least May 2003 and remains in current editions of the H.264 Standard. Any ASUS device that includes a decoder that practices the functionality in any of these editions of the H.264 Standard ("H.264 Decoder) practices the decoding claims of the '321 Patent.

Nokia contends each of the following limitations is met literally, and, to the extent a limitation is not met literally, it is met under the doctrine of equivalents.[2]

[1] See, e.g., https://www.asus.com/us/laptops/for-home/everyday-use/asus-vivobook-pro-15-oled-q543/techspec/; https://www.intel.com/content/www/us/en/products/sku/236849/intel-core-ultra-9-processor-185h-24m-cache-up-to-5-10-ghz/specifications.html; https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new.

[2] This claim chart is based on the information currently available to Nokia and is intended to be exemplary in nature. Nokia reserves all rights to update and elaborate its infringement positions, including as Nokia obtains additional information during discovery.

U.S. Patent No. 8,050,321	ASUS Accused Products
8. [A] A method for decoding a compressed video sequence,	Each of the Accused Products, such as the Asus Q543MV, performs a method for decoding a compressed video sequence. For example, and without limitation, the Asus Q543MV uses hardware-accelerated video decoding and includes an NVIDIA GeForce RTX 4060 Laptop graphics processing unit ("GPU") and an Intel Core Ultra 9 Processor 185H.  Source: https://www.asus.com/us/laptops/for-home/everyday-use/asus-vivobook-pro-15-oled-q543/techspec/ (last accessed March 6, 2025).  Source: https://www.intel.com/content/www/us/en/products/sku/236849/intel-core-ultra-9-processor-185h-24m-cache-up-to-5-10-ghz/specifications.html (last accessed March 6, 2025) (specifications for Intel Core Ultra 9 185H).

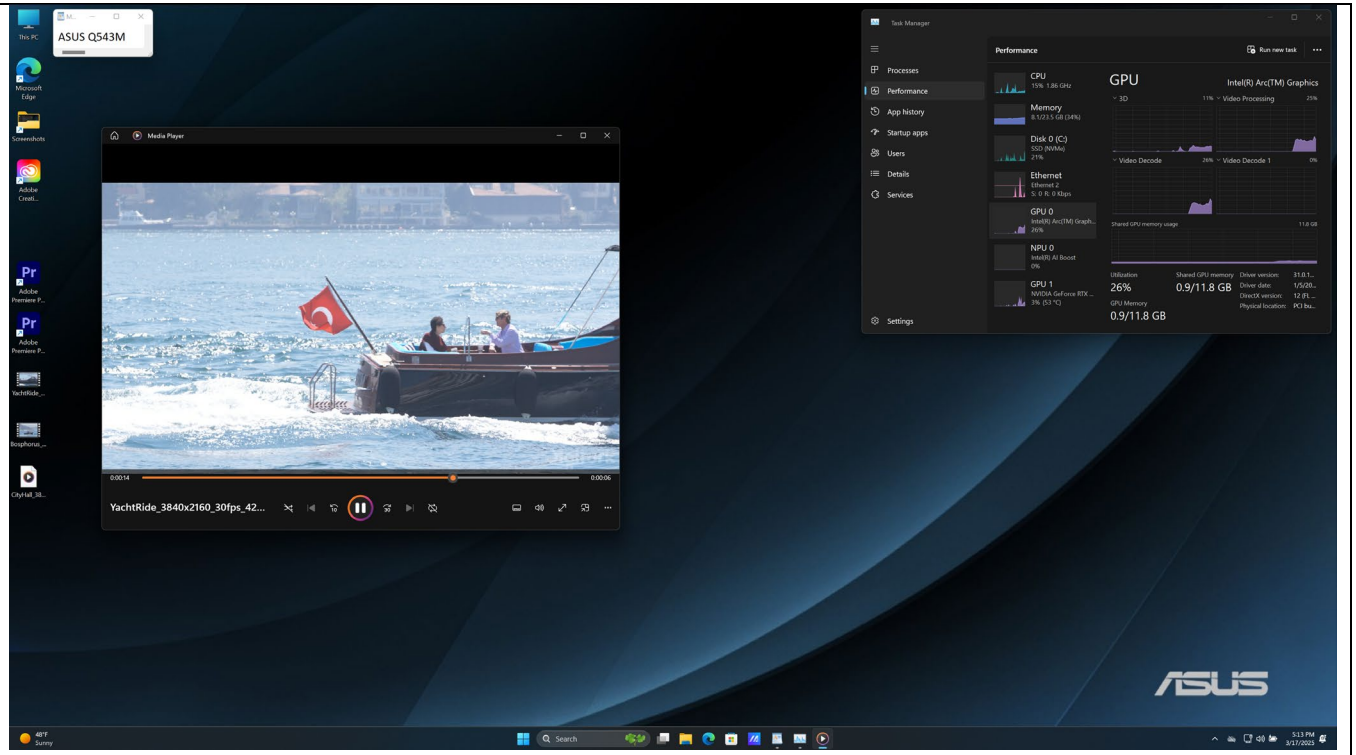
BOARD	FAMILY	NVENC Generation	Desktop/ Mobile	# OF CHIPS	Total # of NVENC	Max # of concurrent sessions	H.264 (AVCHD) YUV 4:2:0	H.264 (AVCHD) YUV 4:2:2	H.264 (AVCHD) YUV 4:4:4	H.264 (AVCHD) Lossless	H.265 (HEVC) 4K YUV 4:2:0	H.265 (HEVC) YUV 4:2:2	H.265 (HEVC) 4K YUV
GeForce RTX 4060 Laptop	Ada Lovelace	8th Gen	M	1	1	8	YES	NO	YES	YES	YES	NO	YES
GeForce RTX 4060	Ada Lovelace	8th Gen	D	1	1	8	YES	NO	YES	YES	YES	NO	YES

BOARD	FAMILY	NVDEC Generation	Desktop/ Mobile	# OF CHIPS	Total # of NVDEC	MPEG-1	MPEG-2	VC-1	VP8	VP9 4:2:0			H.264 (AVCHD) 4:2:0	
										8 Bit	10 Bit	12 Bit	8 Bit	10 Bit
GeForce RTX 4060 Laptop	Ada Lovelace	5th Gen	M	1	1	YES	YES	YES	YES	YES	YES	YES	YES	NO

H.265 (HEVC) 4:2:0			H.265 (HEVC) 4:2:2		H.265 (HEVC) 4:4:4			AV1	
8 Bit	10 Bit	12 Bit	8 Bit	10 Bit	8 Bit	10 Bit	12 Bit	8 Bit	10 Bit
YES	YES	YES	NO	NO	YES	YES	YES	YES	YES

Source: https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new (last accessed March 6, 2025) (row for 4060 Laptop GPU).

For example, an ASUS Q543MV was used to playback an H.264-compliant video.



Source: Screenshot of H.264-compliant video playback on ASUS Q543MV.

For example, and without limitation, the H.264 Standard specifies the following regarding the decoding process. The following specifications provide further evidence of how each of the Accused Products operates:

> **3 Definitions**
> For the purposes of this Recommendation | International Standard, the following definitions apply.
> . . .
> **3.30 coded video sequence**: A sequence of *access units* that consists, in decoding order, of an *IDR access unit* followed by zero or more non-IDR *access units* including all subsequent *access units* up to but not including any subsequent *IDR access unit*.
> . . .

	3.37 decoded picture: A *decoded picture* is derived by decoding a *coded picture*. A *decoded picture* is either a decoded *frame*, or a decoded *field*. A decoded *field* is either a decoded *top field* or a decoded *bottom field*. . . . **3.39 decoder:** An embodiment of a *decoding process*. **3.40 decoding order:** The order in which *syntax elements* are processed by the *decoding process*. **3.41 decoding process:** The process specified in this Recommendation | International Standard that reads a *bitstream* and derives *decoded pictures* from it. . . . (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at pp. 4 – 6).
[B] the method comprising: decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence;	Each of the Accused Products, such as the Asus Q543MV, performs a method of decoding a compressed video sequence, the method comprising: decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence. For example, and without limitation, the H.264 Standard specifies decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence. For example, according to the H.264 Standard, a decoder decodes each coded video sequence starting with an instantaneous decoding refresh (IDR) picture, which is followed by zero or more non-IDR pictures in decoding order. An IDR picture is a coded picture in which all slices are I or SI (intra coded) slices and therefore can be independently decoded without prediction from any other decoded picture. Furthermore, after decoding of an IDR picture, all subsequent coded pictures in decoding order in the coded video sequence can be independently decoded without inter prediction from any picture decoded prior to the IDR picture in question. This means that, for any given coded video sequence after an IDR picture, motion-compensated prediction references only refer to coded pictures within the independent video sequence.

For example, an H.264 Decoder receives an indication of an IDR picture in the bitstream is indicated, for example, as a value of nal_unit_type = 5 in the NAL Unit syntax.

The following specifications provide further evidence of how the at least one H.264 Decoder implemented in each of the Accused Products operates:

> **3 Definitions**
> For the purposes of this Recommendation | International Standard, the following definitions apply.
> **3.1 access unit**: A set of *NAL units* always containing exactly one *primary coded picture*. In addition to the *primary coded picture*, an access unit may also contain one or more *redundant coded pictures* or other *NAL units* not containing *slices* or *slice data partitions* of a *coded picture*. The decoding of an access unit always results in a *decoded picture*.
> . . .
> **3.27 coded picture**: A *coded representation* of a *picture*. A coded picture may be either a *coded field* or a *coded frame*. Coded picture is a collective term referring to a *primary coded picture* or a *redundant coded picture*, but not to both together.
> . . .
> **3.30 coded video sequence**: A sequence of *access units* that consists, in decoding order, of an *IDR access unit* followed by zero or more non-IDR *access units* including all subsequent *access units* up to but not including any subsequent *IDR access unit*.
> . . .
> **3.62 instantaneous decoding refresh (IDR) picture**: A *coded picture* in which all *slices* are *I* or *SI slices* that causes the *decoding process* to mark all *reference pictures* as "unused for reference" immediately after decoding the IDR picture. After the decoding of an IDR picture all following *coded pictures* in *decoding order* can be decoded without *inter prediction* from any *picture* decoded prior to the IDR picture. The first *picture* of each *coded video sequence* is an IDR picture.
> . . .
> **3.87 NAL unit**: A syntax structure containing an indication of the type of data to follow and *bytes* containing that data . . .
> . . .
> **3.109 primary coded picture**: The coded representation of a *picture* to be used by the *decoding process* for a bitstream conforming to this Recommendation | International Standard. The primary coded picture contains all *macroblocks* of the *picture.* The only *pictures* that have a normative effect on the *decoding process* are primary coded pictures. See also *redundant coded picture.*

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at pp. 4, 6 – 9).

7.3.1 NAL unit syntax

nal_unit(NumBytesInNALunit) {	C	Descriptor
forbidden_zero_bit	All	f(1)
nal_ref_idc	All	u(2)
nal_unit_type	All	u(5)
NumBytesInRBSP = 0		
for(i = 1; i < NumBytesInNALunit; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x000003) {		
rbsp_byte[NumBytesInRBSP++]	All	b(8)
rbsp_byte[NumBytesInRBSP++]	All	b(8)
i += 2		
emulation_prevention_three_byte /* equal to 0x03 */	All	f(8)
} else		
rbsp_byte[NumBytesInRBSP++]	All	b(8)
}		
}		

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 38).

7.4.1 NAL unit semantics

. . .

nal_ref_idc not equal to 0 specifies that the content of the NAL unit contains a sequence parameter set or a picture parameter set or a slice of a reference picture or a slice data partition of a reference picture.

nal_ref_idc equal to 0 for a NAL unit containing a slice or slice data partition indicates that the slice or slice data partition is part of a non-reference picture.

. . .

nal_unit_type specifies the type of RBSP data structure contained in the NAL unit as specified in Table 7-1. VCL NAL units are specified as those NAL units having nal_unit_type equal to 1 to 5, inclusive. All remaining NAL units are called non-VCL NAL units.

. . .

<table>
<tr><td></td><td>

Table 7-1 – NAL unit type codes

nal_unit_type	Content of NAL unit and RBSP syntax structure	C
0	Unspecified	
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp( )	2, 3, 4
2	Coded slice data partition A slice_data_partition_a_layer_rbsp( )	2
3	Coded slice data partition B slice_data_partition_b_layer_rbsp( )	3
4	Coded slice data partition C slice_data_partition_c_layer_rbsp( )	4
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp( )	2, 3

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at pp. 56-57).

Further, the evidence cited for claim limitation 8[A] applies to this claim limitation.

</td></tr>
<tr><td>**[C]** starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;</td><td>Each of the Accused Products, such as the Asus Q543MV, performs a method of decoding a compressed video sequence, the method comprising: starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.

For example, and without limitation, the H.264 Standard specifies starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame.

For example, according to the H.264 Standard, each of the Accused Products starts the decoding of each coded video sequence with an instantaneous decoding refresh (IDR) picture, which is followed by zero or more non-IDR pictures in decoding order in the coded video sequence. An IDR picture is a coded picture in which all slices are I or SI (intra coded) slices and therefore each of the Accused Products</td></tr>
</table>

	independently decode IDR pictures without prediction from any other decoded picture. Furthermore, after decoding of an IDR picture, each of the Accused Products can independently decode all subsequent coded pictures in decoding order in the video sequence without inter prediction from any picture decoded prior to the IDR picture in question. **3 Definitions** For the purposes of this Recommendation | International Standard, the following definitions apply. **. . .** **3.30 coded video sequence**: A sequence of *access units* that consists, in decoding order, of an *IDR access unit* followed by zero or more non-IDR *access units* including all subsequent *access units* up to but not including any subsequent *IDR access unit*. **. . .** **3.62 instantaneous decoding refresh (IDR) picture**: A *coded picture* in which all *slices* are *I* or *SI slices* that causes the *decoding process* to mark all *reference pictures* as "unused for reference" immediately after decoding the IDR picture. After the decoding of an IDR picture all following *coded pictures* in *decoding order* can be decoded without *inter prediction* from any *picture* decoded prior to the IDR picture. The first *picture* of each *coded video sequence* is an IDR picture. **. . .** **3.87 NAL unit**: A syntax structure containing an indication of the type of data to follow and *bytes* containing that data **. . .** **. . .** (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at pp. 6 – 8). **7.4.1.2 Order of NAL units and association to coded pictures, access units, and video sequences** This subclause specifies constraints on the order of NAL units in the bitstream **. . .** Decoders conforming to this Recommendation | International Standard shall be capable of receiving NAL units and their syntax elements in decoding order. **. . .** **7.4.1.2.2 Order of access units and association to coded video sequences** A bitstream conforming to this Recommendation | International Standard consists of one or more coded video sequences. A coded video sequence consists of one or more access units. The order of NAL units and coded pictures and their association to access units is described in subclause 7.4.1.2.3.

	The first access unit of each coded video sequence is an IDR access unit. All subsequent access units in the coded video sequence are non-IDR access units. (ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at pp. 56-57). Further, the evidence cited for claim limitations 8[A-B] applies to this claim limitation.
[D] decoding identifier values for image frames according to a numbering scheme; and	Each of the Accused Products, such as the Asus Q543MV, performs a method of decoding a compressed video sequence, the method comprising: decoding identifier values for image frames according to a numbering scheme. For example, and without limitation, as further evidence of how each of the Accused Products operates, the H.264 Standard specifies decoding identifier values for image frames according to a numbering scheme. For example, according to the H.264 Standard, each of the Accused Products decodes a syntax element pic_order_cnt_lsb in the slice header syntax as a numbered identifier for pictures:

7.3.3 Slice header syntax

slice_header( ) {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
if(separate_colour_plane_flag == 1)		
colour_plane_id	2	u(2)
frame_num	2	u(v)
if(!frame_mbs_only_flag) {		
field_pic_flag	2	u(1)
if(field_pic_flag)		
bottom_field_flag	2	u(1)
}		
if(nal_unit_type == 5)		
idr_pic_id	2	ue(v)
if(pic_order_cnt_type == 0) {		
pic_order_cnt_lsb	2	u(v)
if(pic_order_present_flag && !field_pic_flag)		
delta_pic_order_cnt_bottom	2	se(v)

. . .

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 45).

Picture Order Count (picture_order_cnt_lsb) is an identifier for pictures:

pic_order_cnt_lsb specifies the picture order count modulo MaxPicOrderCntLsb for the top field of a coded frame or for a coded field. The size of the pic_order_cnt_lsb syntax element is log2_max_pic_order_cnt_lsb_minus4 + 4 bits. The value of the pic_order_cnt_lsb shall be in the range of 0 to MaxPicOrderCntLsb – 1, inclusive.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 77).

8.2 Slice decoding process

8.2.1 Decoding process for picture order count

Outputs of this process are TopFieldOrderCnt (if applicable) and BottomFieldOrderCnt (if applicable).

Picture order counts are used to determine initial picture orderings for reference pictures in the decoding of B slices (see subclauses 8.2.4.2.3 and 8.2.4.2.4), to represent picture order differences between frames or fields for motion vector derivation in temporal direct mode (see subclause 8.4.1.2.3), for implicit mode weighted prediction in B slices (see subclause 8.4.2.3.2), and for decoder conformance checking (see subclause C.4).

Picture order count information is derived for every frame, field (whether decoded from a coded field or as a part of a decoded frame), or complementary field pair as follows:

– Each coded frame is associated with two picture order counts, called TopFieldOrderCnt and BottomFieldOrderCnt for its top field and bottom field, respectively.

– Each coded field is associated with a picture order count, called TopFieldOrderCnt for a coded top field and BottomFieldOrderCnt for a bottom field.

– Each complementary field pair is associated with two picture order counts, which are the TopFieldOrderCnt for its coded top field and the BottomFieldOrderCnt for its coded bottom field, respectively.

TopFieldOrderCnt and BottomFieldOrderCnt indicate the picture order of the corresponding top field or bottom field relative to the first output field of the previous IDR picture or the previous reference picture including a memory_management_control_operation equal to 5 in decoding order.

TopFieldOrderCnt and BottomFieldOrderCnt are derived by invoking one of the decoding processes for picture order count type 0, 1, and 2 in subclauses 8.2.1.1, 8.2.1.2, and 8.2.1.3, respectively. When the current picture includes a memory management control operation equal to 5, after the decoding of the current picture, tempPicOrderCnt is set equal to PicOrderCnt(CurrPic), TopFieldOrderCnt of the current picture (if any) is set equal to TopFieldOrderCnt - tempPicOrderCnt, and BottomFieldOrderCnt of the current picture (if any) is set equal to BottomFieldOrderCnt - tempPicOrderCnt.

The bitstream shall not contain data that results in Min(TopFieldOrderCnt, BottomFieldOrderCnt) not equal to 0 for a coded IDR frame, TopFieldOrderCnt not equal to 0 for a coded IDR top field, or BottomFieldOrderCnt not equal to 0 for a coded IDR bottom field. Thus, at least one of TopFieldOrderCnt and BottomFieldOrderCnt shall be equal to 0 for the fields of a coded IDR frame.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 98).

8.2.1.1 Decoding process for picture order count type 0

This process is invoked when pic_order_cnt_type is equal to 0.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 99).

<table>
<tr><td></td><td>

8.2.1.2 Decoding process for picture order count type 1

This process is invoked when pic_order_cnt_type is equal to 1.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 100).

8.2.1.3 Decoding process for picture order count type 2

This process is invoked when pic_order_cnt_type is equal to 2.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 101).

For example, according to the H.264 Standard, an H.264 Decoder decodes a syntax element **frame_num** in the slice header syntax of as a numbered identifier for pictures.

7.3.3 Slice header syntax

slice_header( ) {	C	Descriptor
first_mb_in_slice	2	ue(v)
slice_type	2	ue(v)
pic_parameter_set_id	2	ue(v)
frame_num	2	u(v)

. . .

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 45).

7.4.3 Slice header semantics

. . .

frame_num is used as an identifier for pictures . . .

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 76).

Further, the evidence cited for claim limitations 8[A-C] applies to this claim limitation.

</td></tr>
<tr><td>**[E]** resetting the identifier value for the indicated first image frame of the independent sequence.</td><td>Each of the Accused Products, such as the Asus Q543MV, performs a method of decoding a compressed video sequence, the method comprising: resetting the identifier value for the indicated first image frame of the independent sequence.</td></tr>
</table>

	For example, and without limitation, as further evidence of how each of the Accused Products operates, The H.264 Standard specifies resetting of the identifier value for the indicated first image frame of the independent sequence. For example, each of the Accused Products resets prevPicOrderCntMsb and prevPicOrderCntLsb to 0 when it decodes an IDR picture at the beginning of an independent sequence.

8.2 Slice decoding process

8.2.1 Decoding process for picture order count

Outputs of this process are TopFieldOrderCnt (if applicable) and BottomFieldOrderCnt (if applicable).

Picture order counts are used to determine initial picture orderings for reference pictures in the decoding of B slices (see subclauses 8.2.4.2.3 and 8.2.4.2.4), to represent picture order differences between frames or fields for motion vector derivation in temporal direct mode (see subclause 8.4.1.2.3), for implicit mode weighted prediction in B slices (see subclause 8.4.2.3.2), and for decoder conformance checking (see subclause C.4).

Picture order count information is derived for every frame, field (whether decoded from a coded field or as a part of a decoded frame), or complementary field pair as follows:

– Each coded frame is associated with two picture order counts, called TopFieldOrderCnt and BottomFieldOrderCnt for its top field and bottom field, respectively.

– Each coded field is associated with a picture order count, called TopFieldOrderCnt for a coded top field and BottomFieldOrderCnt for a bottom field.

– Each complementary field pair is associated with two picture order counts, which are the TopFieldOrderCnt for its coded top field and the BottomFieldOrderCnt for its coded bottom field, respectively.

TopFieldOrderCnt and BottomFieldOrderCnt indicate the picture order of the corresponding top field or bottom field relative to the first output field of the previous IDR picture or the previous reference picture including a memory_management_control_operation equal to 5 in decoding order.

TopFieldOrderCnt and BottomFieldOrderCnt are derived by invoking one of the decoding processes for picture order count type 0, 1, and 2 in subclauses 8.2.1.1, 8.2.1.2, and 8.2.1.3, respectively. When the current picture includes a memory management control operation equal to 5, after the decoding of the current picture, tempPicOrderCnt is set equal to PicOrderCnt(CurrPic), TopFieldOrderCnt of the current picture (if any) is set equal to TopFieldOrderCnt - tempPicOrderCnt, and BottomFieldOrderCnt of the current picture (if any) is set equal to BottomFieldOrderCnt - tempPicOrderCnt.

The bitstream shall not contain data that results in Min(TopFieldOrderCnt, BottomFieldOrderCnt) not equal to 0 for a coded IDR frame, TopFieldOrderCnt not equal to 0 for a coded IDR top field, or BottomFieldOrderCnt not equal to 0 for a coded IDR bottom field. Thus, at least one of TopFieldOrderCnt and BottomFieldOrderCnt shall be equal to 0 for the fields of a coded IDR frame.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 98).

	8.2.1.1 Decoding process for picture order count type 0 This process is invoked when pic_order_cnt_type is equal to 0. Input to this process is PicOrderCntMsb of the previous reference picture in decoding order as specified in this subclause. Outputs of this process are either or both TopFieldOrderCnt or BottomFieldOrderCnt. The variables prevPicOrderCntMsb and prevPicOrderCntLsb are derived as follows. – If the current picture is an IDR picture, prevPicOrderCntMsb is set equal to 0 and prevPicOrderCntLsb is set equal to 0. – Otherwise (the current picture is not an IDR picture), the following applies. – If the previous reference picture in decoding order included a memory_management_control_operation equal to 5, the following applies. – If the previous reference picture in decoding order is not a bottom field, prevPicOrderCntMsb is set equal to 0 and prevPicOrderCntLsb is set equal to the value of TopFieldOrderCnt for the previous reference picture in decoding order. – Otherwise (the previous reference picture in decoding order is a bottom field), prevPicOrderCntMsb is set equal to 0 and prevPicOrderCntLsb is set equal to 0. – Otherwise (the previous reference picture in decoding order did not include a memory_management_control_operation equal to 5), prevPicOrderCntMsb is set equal to PicOrderCntMsb of the previous reference picture in decoding order and prevPicOrderCntLsb is set equal to the value of pic_order_cnt_lsb of the previous reference picture in decoding order.

PicOrderCntMsb of the current picture is derived as follows:

```
if( ( pic_order_cnt_lsb < prevPicOrderCntLsb ) &&
    ( ( prevPicOrderCntLsb – pic_order_cnt_lsb ) >= ( MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb + MaxPicOrderCntLsb                    (8-3)
else if( ( pic_order_cnt_lsb > prevPicOrderCntLsb ) &&
        ( ( pic_order_cnt_lsb – prevPicOrderCntLsb ) > ( MaxPicOrderCntLsb / 2 ) ) )
    PicOrderCntMsb = prevPicOrderCntMsb – MaxPicOrderCntLsb
else
    PicOrderCntMsb = prevPicOrderCntMsb
```

When the current picture is not a bottom field, TopFieldOrderCnt is derived as follows:

```
if( !field_pic_flag || !bottom_field_flag )
    TopFieldOrderCnt = PicOrderCntMsb + pic_order_cnt_lsb                      (8-4)
```

When the current picture is not a top field, BottomFieldOrderCnt is derived as follows:

```
if( !field_pic_flag )
    BottomFieldOrderCnt = TopFieldOrderCnt + delta_pic_order_cnt_bottom
else if( bottom_field_flag )                                                   (8-5)
    BottomFieldOrderCnt = PicOrderCntMsb + pic_order_cnt_lsb
```

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at pp. 99-100).

8.2.1.2 Decoding process for picture order count type 1

This process is invoked when pic_order_cnt_type is equal to 1.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 100).

8.2.1.3 Decoding process for picture order count type 2

This process is invoked when pic_order_cnt_type is equal to 2.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 101).

Further, 8.2.1.1 applies when NAL unit type = 5:

8.1 NAL unit decoding process

Inputs to this process are NAL units.

Outputs of this process are the RBSP syntax structures encapsulated within the NAL units.

The decoding process for each NAL unit extracts the RBSP syntax structure from the NAL unit and then operates the decoding processes specified for the RBSP syntax structure in the NAL unit as follows.

Subclause 8.2 describes the decoding process for NAL units with nal_unit_type equal to 1 through 5.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 97).

For example, an H.264 Decoder resets frame_num to 0 when it decodes an IDR picture at the beginning of an independent sequence.

7.4.3 Slice header semantics

. . .

frame_num is used as an identifier for pictures . . .

. . .

The value of frame_num is constrained as follows.

– If the current picture is an IDR picture, frame_num shall be equal to 0.

(ITU-T Rec. H.264 (03/2005) Advanced video coding for generic audiovisual services, at p. 76).

Further, the evidence cited for claim limitations 8[A-D] applies to this claim limitation.